HEWLETT PACKARD

Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

00071-90019 English Printed in U.S.A. 10/83

HP-71 Quick Reference Guide

Contents

CALC Mode Operations

[ATTN]	Clears the display.
[f][BACK]	Back-spaces one character. Also restores the expression before the last evaluation.
[f][CALC]	Switches the computer between CALC mode and BASIC mode.
[g][CMDS]	Enters and exits the command stack.
[f][CONT]	Same as [END LINE].
[END LINE]	Returns the value of an expression. If no expression is in the display, then returns the value of **RES**.
[g][ERRM]	Displays the last error or warning message.
[f][I/R]	Switches the command stack cursor between insert and replace.
[f][LC]	Switches between uppercase and lowercase lock.

f -LINE	In command stack, erases from cursor to the end of the line. When not in command stack, erases the entire line.
f OFF	Turns the computer off.
f RES	Displays **RES**, which represents the result of the most recently evaluated expression.
f SST	Evaluates the right-most operation or function of an expression.
f USER	Activates and deactivates the User keyboard.
g 1 USER	Activates and deactivates the User keyboard for one shifted or unshifted keystroke.
f VIEW	Displays the key definition of the next pressed key.
▲, ▼	Scrolls through the command stack. Also enters and exits the command stack.
◀, ▶	Moves the cursor horizontally through a command stack line. If the line is longer than display, then the line scrolls horizontally through the display.
g ⤒, g ⤓	Displays the top and bottom lines in the command stack, respectively. Also enters the command stack.
g ⇤, g ⇥	Moves the cursor to the left-most character and to the right-most character in a command stack line, respectively. If the line is longer than the display, then the line scrolls horizontally through the display.
key definitions	Can be used if they are not direct execute (:) definitions.
↵	Shows that an expression in the command stack was evaluated.
()	Specifies result of the previously evaluated expression.

Operator Summary

Binary Operator	Operation
+	Addition.
−	Subtraction.
*	Multiplication.
/	Division.
^	Exponentiation.
DIV or \	Integer Division (\ is **CHR$(92)**).
%	The operation *x* **%** *y* returns *x* percent of *y*.
&	Concatenates two strings.
=	Equal to?
>	Greater than?
>=	Greater than or equal to?
<	Less than?
<=	Less than or equal to?
<>	Less than or greater than?
#	Not equal to?
?	Unordered?
AND	Both expressions true (that is, nonzero)?
OR	Either expression true?
EXOR	One or the other expression true, but not both?

Unary Operator	Operation
−	Reverses the sign of an expression.
+	Identity operator.
NOT	Is the expression false (that is, zero)?

Precedence of Operators

The table below lists HP-71 operators in their order of precedence. Where an expression contains two or more operators having the same level of precedence, those operators will be evaluated in the left-to-right order in which they occur within the expression.

Performed first.
(...) (Nested parentheses are evaluated from the inside out.)
Functions (such as **SIN**, **RND**, etc.)
^
unary +, unary −, **NOT**
*, \, **DIV**, **%**
+, −, **&**
<, =, >, #, **?**, <=, >=, <>
AND
OR, **EXOR**
Performed last.

Syntax Symbols

The Function Summary and Statement Summary use the following symbols to define syntax:

BOLD	Keywords and item separators shown in bold and blue must be typed as shown.
italic	Parameters are shown in italic typeface.
[ ]	Parameters enclosed in square brackets are optional.
...	Parameters can be repeated
/	Keywords separated by a slash are interchangeable.

Also, keyword headings with no parameters are typed as shown. Stacked items are interchangeable.

Function Summary

ABS(*x*)	Returns the absolute value of *x*.
ACOS(*x*) or ACS(*x*)	Returns the inverse cosine of *x*.
ADDR$(*file specifier*)	Returns the hexadecimal address of the start of the specified file.
AF [(*x*)]	Returns the clock accuracy factor and optionally sets the factor.
ANGLE(*x*,*y*)	Returns the polar angle determined by the *x*, *y* coordinate pair.
ASIN(*x*) or ASN(*x*)	Returns the inverse sine of *x*.
ATAN(*x*) or ATN(*x*)	Returns the inverse tangent of *x*.
CAT$(*x* [,*device*])	Returns the catalog entry for the file indicated by *x* in the optionally specified device.
CEIL(*x*)	Returns the smallest integer greater than or equal to *x*.
CHARSET$	Returns a string representing the current alternate character set.
CHR$(*x*)	Returns the character whose character code is *x*.
CLASS(*x*)	Returns a value indicating the numeric class of *x*.
CORR(*x*,*y*)	Returns the correlation coefficient between variables *x* and *y* in the current statistical array.
COS(*x*)	Returns the cosine of *x*.

DATE	Returns the date as an integer (*YYDDD*).
DATE$	Returns the date as a string (*YY/MM/DD*).
DEG(*x*)	Converts *x* from radians to degrees.
DISP$	Returns a string containing all readable characters in the display.
DTH$(*x*)	Converts *x* from a decimal value to a string representing its five-digit hexadecimal value.
DVZ	Returns the divide-by-zero flag number (−7).
EPS	Returns the smallest, positive, normalized number that the HP-71 can represent (1.0E−499).
ERRL	Returns the line number where the most recent program error or warning occurred.
ERRM$	Returns the text of the most recent error or warning message.
ERRN	Returns the number of the most recent error or warning.
EXP(*x*)	Returns the value of *e*^*x*.
EXPM1(*x*)	Returns the value of *e*^*x*−1.
EXPONENT(*x*)	Returns the exponent of the normalized equivalent of *x*.
FACT(*x*)	Returns the factorial of *x*.
FLAG(*flag*[,*new value*])	Returns the current value (0 or 1) of the specified flag and optionally sets or clears the flag.

FLOOR(*x*)	Returns the greatest integer less than or equal to *x*.
FN *variable name* [(*parameters*)]	Returns the value of the specified user-defined function.
FP(*x*)	Returns the fractional part of *x*.
GDISP$	Returns a 132-character string representing the bit pattern in the display.
HTD(*hex value*)	Converts a string representing a five-digit hexadecimal number into a decimal value.
INF	Returns the computer representation of positive infinity.
INT(*x*)	Returns the greatest integer that is less than or equal to *x*.
INX	Returns the number of the inexact result flag (−4).
IP(*x*)	Returns the integer part of *x*.
IVL	Returns the number of the invalid operation flag (−8).
KEY$	Removes the oldest shifted or unshifted keystroke from the key buffer and returns a string representing it.
KEYDEF$(*key*)	Returns the redefined value of *key*.
KEYDOWN [(*key*)]	Returns 1 if a key is pressed, otherwise returns 0.
LEN(*string*)	Returns the length of *string*.
LOG(*x*) or LN(*x*)	Returns the natural logarithm (base *e*) of *x*.

LOGP1(*x*)	Returns the value of ln(1+*x*).
LOG10(*x*) or LGT(*x*)	Returns the common logarithm (base 10) of *x*.
MAX(*x*,*y*)	Returns *x* or *y*, whichever is greater.
MAXREAL	Returns the maximum positive finite number that the computer can represent (9.99999999999E499).
MEAN [(*x*)]	Returns the sample mean of the specified variable in the current statistical array. The default for *x* is 1.
MEM [(*port specifier*)]	Returns the amount of unused memory in either main RAM or a specified port.
MIN(*x*,*y*)	Returns either *x* or *y*, whichever is less.
MINREAL	Returns the smallest positive number that the computer can represent (0.00000000001E−499).
MOD(*x*,*y*)	Returns the value of *x* modulo *y*, which is calculated by the expression $x-y\times$**INT**(x/y).
NAN	Returns a signaling NaN.
NUM(*string*)	Returns the character code of the first character in *string*.
OVF	Returns the number of the overflow flag (−6).
PEEK$(*hex address*,*x*)	Returns the contents of *x* nibbles of memory beginning at the specified address.

PI	Returns the 12-digit value of π.
POS(*string*,*substring* [,*start*])	Returns the position of the specified substring in the specified string. The default value for *start* is 1.
PREDV(*x*)	Returns the predicted value of a dependent variable based on the value, *x*, given for the independent variable.
RAD(*x*)	Converts *x* from degrees to radians.
RED(*x*,*y*)	Returns a remainder defined by $x-y\times n$, where n is the nearest integer to x/y.
RES	Returns the value of the most recently evaluated numeric expression.
RMD(*x*,*y*)	Returns the remainder defined by $x-y\times$**IP**(x/y).
RND	Returns a random number and updates the random number seed.
SDEV[(*x*)]	Returns the sample standard deviation for the specified variable in the current statistical array. The default value for *x* is 1.
SGN(*x*)	Returns a value indicating the sign of *x* (−1 if $x<0$, 0 if $x=0$, 1 if $x>0$).
SIN(*x*)	Returns the sine of *x*.
SQRT(*x*) or SQR(*x*)	Returns the square root of *x*.
STR$(*x*)	Returns a string representation of the value of *x*.

TAN(*x*)	Returns the tangent of *x*.
TIME	Returns the number of seconds since midnight.
TIME$	Returns the time of day as a string in 24-hour format (*HH*:*MM*:*SS*).
TOTAL [(*x*)]	Returns the total of the specified variable in the current statistical array. The default value for *x* is 1.
TRAP(*flag*[,*new value*])	Returns the trap value of *flag* and optionally sets a *new value*.
UNF	Returns the number of the underflow flag (−5).
UPRC$(*string*)	Returns a string with all characters converted to uppercase.
VAL(*string*)	Returns the value of a string as if it were a numeric expression.
VER$	Returns a string that indicates the version of the computer's operating system and any plug-in applications ROMs.

Statement Summary

This summary lists the statements used by the HP-71, and provides for each statement a short description, a brief syntax, and examples of use.

A

ADD [*coordinate* [,*coordinate*...]]

Adds the *coordinates* of a data point to the summary statistics in the current statistical array.

```
ADD                          ADD 1,34.87365,A,B
ADD V1, V2, V3, V4, V5
```


ADJABS *seconds*
ADJABS *adjustment string*

Performs an *absolute* adjust on the system clock.

```
ADJABS 124.7                 ADJABS "+00:05:45"
ADJABS −3300                 ADJABS "−00:02:14"
ADJABS "00:03:35"            ADJABS N
```

ADJUST *seconds*
ADJUST *adjustment string*

Simultaneously changes the clock time and specifies a clock speed correction factor.

```
ADJUST 124.7                 ADJUST "+00:05:45"
ADJUST −3300                 ADJUST "−00:02:14"
ADJUST "00:03:35"            ADJUST A$
```

ASSIGN # *channel number* TO *file specifier*

Associates a symbolic *channel number* with a specified file and opens that file.

```
ASSIGN #1 TO FILE1
ASSIGN #L TO FILE2:PORT(2)
ASSIGN #6 TO *
```

AUTO [*start line number* [, *increment*]]

Enables automatic line numbering for editing the current file. (*Not programmable.*)

```
AUTO 100, 25      AUTO 25      AUTO
```

B

BEEP [*frequency* [, *duration*]]

Causes a tone to sound at the specified *frequency* and *duration*. Default *frequency* is 500 Hz and default *duration* is 0.25 seconds.

```
BEEP              BEEP 261,1   BEEP F
```

BEEP OFF/ON

- **BEEP OFF** disables the beeper.
- **BEEP ON** enables the beeper.

BYE

Turns off the HP-71.

C

CALL [*subprogram name* [(*parameters*)] [IN *file specifier*]]

Transfers program execution to a specified *subprogram*, and may also pass *parameters* to that subprogram.

```
CALL
CALL C1$(A,B$,X+Y,"YES")
CALL SETUP1
CALL CONTROL(A1(4),P(),Q(,),D$[3,10]) IN IOSUBS
CALL "EXECUTE"((L), 3, N) IN GENERAL
```

CAT [*file specifier*]

Gives a catalog of file information.

```
CAT TESTPR3        CAT :PORT(1)      CAT A$
CAT ABC:PORT       CAT KEYS          CAT ALL
CAT:MAIN           CAT               CAT CARD
```

CFLAG *flag number* [, *flag number*...]
CFLAG ALL/MATH

Clears the flag specified by each *flag number*, the math exception flags, or all user flags and settable system flags.

CFLAG clears flags as follows:

- **CFLAG ALL** clears all user flags (0 through 63).
- **CFLAG MATH** clears the math exception flags (−8 through −4).
- **CFLAG** clears specified user and system flags. A *flag number* is a numeric expression whose integer-rounded value represents a flag number.

```
CFLAG 1,2,INX      CFLAG MATH        CFLAG ALL
```


CHAIN *file specifier*

Purges the current file, copies the specified file into main RAM, and begins executing that file.

```
CHAIN SEGMENT5          CHAIN TEST2:CARD
CHAIN P$                CHAIN DEMO:PORT(2)
```

CHARSET *string*

Defines the alternate character set.

```
CHARSET CHR$(64)&CHR$(128)&CHR$(126)
 &CHR$(1)&CHR$(2)&CHR$(0)
CHARSET CHARSET$&CHR$(1)&CHR$(128)&CHR$(1)
 &CHR$(128)&CHR$(1)
CHARSET C$
CHARSET ""
```

CLAIM PORT (*port specifier*)

Returns an independent RAM to main RAM status. (*Not programmable.*)

```
CLAIM PORT(0)           CLAIM PORT(A)
```

CLSTAT

Clears (sets to zero) all elements in the current statistical array.

CONT [*statement identifier*]

Continues execution of a suspended program at an optionally specified *statement identifier* (line number or label). (*Not programmable.*)

```
CONT                    CONT SECTION4
CONT 250                CONT "TEST5"
CONT L$
```

CONTRAST *contrast value*

Adjusts the display contrast (viewing angle). The *contrast value* can be any numeric expression. The computer rounds the *contrast value* to an integer in the range 0 through 15. If the specified *contrast value* is less than 0, it is set to 0; if it is greater than 15, it is set to 15.

```
CONTRAST 10
```

COPY [*source file specifier*] [TO *destination file specifier*]

Creates a new *destination file* and copies information from an existing *source file* to the new file.

```
COPY TO DEMO1                    COPY GAMES:PORT
COPY FILE1 TO FILE2              COPY CARD
COPY TO CARD                     COPY A$ TO CARD
COPY K1:PORT(2) TO KEYS
COPY CARD TO GRAPH
COPY "TEST" TO TEST5:PCRD
COPY KEYS TO KEYASN:PORT(1)
```

CREATE *file type file specifier* [,*file size* [,*record length*]]

Creates a data file in main RAM or in an indendent RAM.

```
CREATE TEXT MEMO245
CREATE DATA A$,20,50
CREATE TEXT "FILE6:PORT(1)",2000
CREATE SDATA FILE41C,200
```


D

DATA *data item* [, *data item...*]

Contains data that can be read by **READ** statements.

```
DATA 123,456,"Hello"
DATA 'This is a string'&' expression'
DATA PI*2,FGH'QQQ'
```

DEF FN *variable name* [(*parameters*)] [= *expression*]

Indicates the beginning of a user-defined function definition. If *expression* is present, a single statement user-defined function is defined. If *expression* is not present, a multistatement user-defined function is defined.

```
DEF FNF(X,Y,Z)= (X*3+Y*4)/Z
DEF FNA(I,J,K,B$)
DEF FNT @ T=X @ X=Y @ Y=T @ END DEF
DEF FNF$(A,B,C$)= C$[A,B]
DEF FNL$[128](T$,R$,X)
DEF FNM5
```

[DEF]KEY *key name* [, *string* [*assignment type*]]

Assigns a character string to the specified key.

```
DEF KEY "#5","GOSUB 134"
KEY "C",A$;
DEF KEY E$,"RUN 200":
KEY "#64",'RUN "SUB1"'
DEF KEY "g1","CALL SUB1(I)"
DEF KEY "FQ","CALL PMT"
DEF KEY "A"
```

The following table describes the three types of key assignments:

Assignment Key	Punctuation Used To Define Key Assignment Type	Result of Pressing the Assigned Key
Typing Aid	Semicolon (;).	Displays the assigned string as though had you manually typed it.
Direct Execution	Colon (:).	Executes the assigned string without altering the display.
Immediate Execution	None (default).	Displays the assigned string then executes it.

DEFAULT OFF/ON/EXTEND

DEFAULT sets the math exception traps to specific values.

Math Exception Trap Values Set By DEFAULT

Default	Math Exception Flag IVL	DVZ	OVF	UNF	INX
DEFAULT OFF	0	0	0	0	1
DEFAULT ON	1	1	1	1	1
DEFAULT EXTEND	1	2	2	2	2

DEGREES

Sets the unit of measure for expressing angles to degrees. **DEGREES** is a short form of the **OPTION ANGLE DEGREES** statement.

DELAY *line rate* [*character rate*]

Sets the rate at which lines and characters within a line scroll across the display.

```
DELAY 1                    DELAY 0,INF
```

DELETE *start line number* [*end line number*]
DELETE ALL

Deletes one or more program lines from the current file. (*Not programmable.*)

```
DELETE 10                  DELETE 100,200
```

DESTROY *variable* [*variable...*]
DESTROY ALL

Deletes either specific or all variables and arrays from memory.

```
DESTROY X                  DESTROY A,R$,M
```

DIM *item* [*item...*]

Allocates memory for string or **REAL** variables and arrays. An *item* can be:

- *numeric variable* [(*dimension 1* [, *dimension 2*])]
- *string variable* [(*dimension*)] [[*length*]]

```
DIM C$(25)[30]  DIM Z2, W(4,4)      DIM A$(12)
DIM W9(7,3)     DIM G$[50]          DIM Z
```

DISP [*display list*]

Displays numeric and string data.

```
DISP
DISP 'Hello ';A$;'.','Welcome'
DISP I;TAB(I+5);'*'
A$,'DATE: ',D$
```

DISP USING *line number* [; *display list*]
DISP USING *format string* [; *display list*]

Displays the items in the *display list* according to a user-specified *format string* that is either included in the statement or located at the line specified by line *number*.

```
DISP USING 200; X, 345.99
DISP USING S$; A$, 1.25+X
DISP USING "#,K,2(XX,3D.D)";
 "Output=",Y,2.34E2
DISP USING "2X,'hello',X,8A,'!'"; N$
DISP USING '8A,5X,"$"3D.DD'; "Profit=", P
```

DROP [*coordinate* [, *coordinate* ...]]

Removes the coordinates of a data point from the data set represented by the summary statistics in the current statistical array.

```
DROP                    DROP 2,12.3,A,C
```

E

EDIT [*file specifier*]

Designates the specified file as the current file. It creates the file if it doesn't already exist. (*Not programmable.*)

```
EDIT FILE6              EDIT
EDIT "STD1"             EDIT A:PORT
```

END [ALL]

- **END** ends a subprogram, user-defined function, or a program.
- **END ALL** terminates program execution, releases local environments, and clears all program control information.

END DEF

Causes a normal return from a called multistatement user-defined function.

END SUB

Causes a normal return from a subprogram.

ENDLINE [*endline string*]

Specifies the end-of-line sequence used by **PRINT** and **PLIST** statements. The *endline string* is limited to three characters.

```
ENDLINE
ENDLINE CHR$(13)&CHR$(10)&CHR$(10)
ENDLINE ""
```

ENG *number of significant digits*

Sets the engineering display format and the *number of significant digits* to be displayed or printed.

```
ENG D
IF X > 9999 THEN ENG 11
```

EXACT

Calibrates the system clock.

F

FETCH [*statement identifier*]

Displays for editing the line in the current file specified by *statement identifier*. The default line is the current line. (*Not programmable.*)

```
FETCH 55                FETCH COMPUTE
FETCH                   FETCH "SECTION2"
```

FETCH KEY *key name*

Displays a specified key assignment for editing. (*Not programmable.*)

```
FETCH KEY "A"           FETCH KEY "fA"
FETCH KEY"#123"
```

FIX *number of significant digits*

Sets the fixed display format and the *number of significant digits* to be displayed or printed.

```
FIX D                          IF X < 1 THEN FIX 11
```

FOR *loop counter* = *initial value* TO *final value* STEP *step size*

NEXT *loop counter*

FOR is used with **NEXT** to define a loop that is repeated until the *loop counter* either exceeds the specified *final value* (using a positive *step size)* or becomes less than the specified *final value* (using a negative *step size).*

```
FOR K = 1 TO 16 @ DISP 16*K @ NEXT K
FOR I = 1 TO 10 @ DISP CAT$(I) @ NEXT I
FOR A1 = 50 TO -100 STEP -.01
```

FREE PORT (*port specifier*)

Switches the main RAM in the specified port to Independent RAM status. (*Not programmable.*)

```
FREE PORT(1.01)                FREE PORT(A)
```

G

GDISP *bit pattern*

Sets the dot pattern in the display according to a specified string.

```
GDISP A$                       GDISP GDISP$[2]
```

GOSUB *statement identifier*

Transfers program execution to the subroutine beginning at the specified statement.

```
GOSUB 100                      GOSUB "PLOT10"
GOSUB G$                       GOSUB COMPUTE6
```


GOTO *statement identifier*

When executed in a program, **GOTO** transfers program execution to the specified statement. When executed from the keyboard, **GOTO** makes the specified line the current line.

```
GOTO 100                       GOTO PRINTVAR
GOTO C$                        GOTO "GRAPHX"
```

I

IF *expression* THEN *statements* [ELSE *statements*]

Provides conditional execution.

```
A$=CAT$(I) @ IF LEN(CAT$(I)) THEN DISP A$[1,8]
IF A=B THEN LABEL5 ELSE GOSUB 100 @ IF A>B
  THEN STOP
IF B THEN DISP B @ STOP ELSE 250
```

IMAGE *unquoted format string*

When used in conjunction with **DISP USING** and **PRINT USING**, controls the format of displayed and printed output.

```
IMAGE M3D.2D,3X,5A
IMAGE "Third Quarter",3(5X,'$'M3ZC3Z.2D)
IMAGE #,K
IMAGE 2(K),XA3"."A/
```

The following table lists the image symbols allowed in a format string. An *n* preceding a symbol indicates that the symbol can be preceded by a multiplier.

IMAGE Symbols

Symbol	Description
,	Separates two field specifiers. (Delimiter.)
#	Suppresses the automatic output of the **ENDLINE** sequence at the end of the output list. If used, # must be the first symbol in the format string and be followed by a delimiter.
n /	**DISP USING** sends a carriage-return, line-feed to the display. **PRINT USING** sends the **ENDLINE** sequence to the print device. (Delimiter.)
n @	Sends a form-feed to the output device. (Delimiter.)
n ( )	Delimits one or more symbols or field specifiers for replicated output. (Delimiter.)
K	Compact field. Displays or prints a number or string in current display format, with no leading or trailing blanks.
H	Same as K, except the European radix (,) is substituted for a decimal point in numeric output.
B	Displays or prints a single character from a character code.
n **X**	Displays or prints a blank. (Editing symbol.)
n "string"	Displays or prints the characters contained in the quoted string. (Editing symbol.)

Symbol	Description
^	Suppression field. Causes the computer to evaluate the corresponding item without displaying or printing the result.
n **A**	Displays or prints character from a string item.
n **D**	Displays or prints one digit and replaces leading zeros with spaces.
n *	Same as for **D**, except leading zeroes are replaced with "*".
n **Z**	Same as for *, except leading zeroes are displayed.
S	Displays or prints a number's sign (+ or −).
M	Displays or prints a number's sign if negative, and a space if positive. No more than one **S** or **M** is allowed per numeric field.
.	Displays or prints a decimal point radix.
R	Displays or prints "," for the European radix. No more than one "." or **R** is allowed per numeric field.
C	Displays or prints a comma as a digit separator.
P	Displays or prints "." for the European digit separator.
E	Displays or prints **E**, with a sign and a three-digit exponent. (Leading zeroes in the exponent are displayed.)

INPUT [*prompt string* [, *default string*]] *input list*

Enables you to assign values to program variables from the keyboard.

```
INPUT "NAME: ",N$;N$
INPUT "Duration,Frequency?";T,F
INPUT X,Y,A$[1,1]
```

INTEGER *item* [, *item...*]

Allocates memory for one or more integer variables and arrays. An *item* can be: *numeric variable* [(*dimension 1* [, *dimension 2*])]

```
INTEGER I                    INTEGER P9(7,3)
INTEGER J2, B(4)
```

L

LC [OFF/ON]

Switches the unshifted and g-shifted letter keys between uppercase and lowercase.

- **LC** switches between uppercase and lowercase lock on the keyboard and switches the state of flag −15.
- **LC OFF** sets the unshifted letter keys to uppercase and clears flag −15.
- **LC ON** sets the unshifted letter keys to lowercase and sets flag −15.

[LET] *variable* = *expression*

Assigns values to variables.

```
LET A=3*B+7                  M(4,3)=4*PI*R^2
A$="STEVEN A."               B7$(I)="TITLES"
Q$[3,5]="MULTIMETER"         LET FNF=Q/P+3
```


LINPUT [*prompt string* [, *default string*]] *string variable*

Assigns an entire line from the display to a *string variable*.

```
LINPUT "City, State: ","Whittier, CA";C$
LINPUT L$
LINPUT "Enter line:";T$
```

LIST [*start line* [, *final line*]]
LIST [*file specifier*] [, *start line* [, *final line*]]
LIST [*start key number* [, *final key number*]]
LIST [*file specifier*] [, *start key number* [, *final key number*]]

Lists the specified BASIC or KEY file.

```
LIST                         LIST MEAN:PORT(2)
LIST A$, 225                 LIST 10
LIST 50,85                   LIST MECH65,100,200
```

LOCK *password*

Specifies a password of up to 8 characters. It provides security against unauthorized use of your HP-71.

```
LOCK "Abc"                   LOCK N$
LOCK ""
```

LR *dependent variable number* , *independent variable number* [, *intercept variable* [, *slope variable*]]

Specifies the current linear regression model and computes the intercept and slope for that model.

```
LR 1,3                       LR 2,1,X,Y
LR 1,2,A(1),A(2)
```

M

MERGE *file specifier* [, *start line* [, *final line*]]
MERGE *file specifier* [, *start key number* [, *final key number*]]

Integrates a portion of the BASIC file you specify into the current file or integrates the KEY file that you specify into the system **keys** file.

```
MERGE A:PORT          MERGE KEYFILE1,100,135
MERGE B$,5
```

N

NAME *file name*

Names the **workfile**.

```
NAME A$&B$      NAME ABC1      NAME "FILE1"
```

O

OFF

Turns off the HP-71. **BYE** also turns the HP-71 off.

OFF ERROR

Disables the active **ON ERROR** condition, thus returning the computer to its default method of reporting errors.

OFF TIMER # *timer number*

Deactivates the corresponding **ON TIMER** # statement.

```
OFF TIMER #1          OFF TIMER #N
OFF TIMER #T1*X/3
```


ON ERROR GOSUB/GOTO *statement identifiers*

Causes the computer to execute the specified subroutine or branch when an error occurs during program execution.

```
ON ERROR GOSUB 1200
ON ERROR GOTO 3000
ON ERROR GOTO "EXIT"
ON ERROR GOSUB E$
ON ERROR GOSUB "ERROR"
ON ERROR GOTO E1$
```

ON *pointer* **GOSUB/GOTO/RESTORE** *statement identifiers*

- **ON...GOSUB** and **ON...GOTO** transfer program execution to the *statement identifier* indicated by the *pointer*. The *pointer* is a numeric expression. For a *pointer* value of 1, the first statement is selected; for a *pointer* value of 2, the second is selected, and so on.
- **ON...RESTORE** resets the **READ** data pointer to the closest **DATA** statement following the selected line number or label.

```
ON X1 GOSUB 100,200,300
ON P2 GOSUB CALC1,CALC2,CALC3,CALC4
ON Y5 GOTO 1000,2000,3000
ON Q1*Y/Z GOTO Q1$,Q2$,Q3$,E$
ON P GOTO COMP1,COMP2,COMP3,COMP4,COMP5
ON D RESTORE 150,250,350
ON E1 RESTORE DATA1,DATA2,DATA3,DATA4,DATA5
ON Q1*Y/Z RESTORE D$(1),D$(2),D$(3),E$
```

ON TIMER # *timer number*, *seconds* **GOSUB/GOTO** *statement identifier*

Enables any one of three timers to interrupt a program at the specified time interval and cause the specified branching to occur.

```
ON TIMER #1,10 GOSUB 2000
ON TIMER #N,I GOSUB "TIMER"
ON TIMER #T*(X1-X2),I*7/32 GOTO T$
```

OPTION ANGLE/BASE/ROUND

Determines the settings for numeric computation.

- **OPTION ANGLE** selects the unit of measure for expressing angles (**RADIANS** or **DEGREES**).
- **OPTION BASE** specifies the subscript lower bound(s) for subsequently dimensioned arrays (either **0** or **1**).
- **OPTION ROUND** selects the round-off setting (**NEAR**, **POS**, **NEG**, or **ZERO**).

```
OPTION ANGLE DEGREES    OPTION BASE 1
OPTION ROUND NEAR
```

P

PAUSE

Suspends program execution.

PLIST [*start line number* [, *final line number*]]
PLIST [*file specifier*] [, *start line number* [, *final line number*]]
PLIST [*start key number* [, *final key number*]]
PLIST [*file specifier*] [, *start key number* [, *final key number*]]

Lists the specified BASIC or KEY file on the print device. (The display is the default print device.)

```
PLIST                    PLIST CAT$(I),10,99
PLIST KEYS               PLIST 100,500
```

POKE *hexadecimal address* , *data*

Writes *data* to memory at the specified *hexadecimal address*.

```
POKE "AF30","56DE"       POKE A$,D$
```

POP

Cancels the pending return of program execution from the current subroutine.

PRINT *print list*

Causes the *print list* items to be sent to the print device.

```
PRINT I;TAB(I+5);"*"
PRINT "Hello ";A$;".","Welcome"
PRINT
```

PRINT USING *line number* [; *print list*]
PRINT USING *image string* [; *print list*]

Causes the *print list* to be sent to the print device in a user-specified format.

```
PRINT USING 200; X, 345.99
PRINT USING S$; A$, 1.25+X
PRINT USING "#,K,2(XX,3D.D)"; "Output=",Y,2.34E2
```

PRINT # *channel number* [, *record*]; *print list*

Writes data items to a data file associated with a specified channel number.

```
PRINT #3; A,B$,X+Y,"SUNDAY"
PRINT #18,20;A,B$,C
PRINT #F; A1(,),D$[3,39],G$()
```

PRIVATE *file specifier*

Permanently limits access to the specified file and restricts changes in its protection.

```
PRIVATE A:PORT(3)        PRIVATE "BCD"
PRIVATE A$
```

PROTECT

Write-protects one track of a magnetic card.

PURGE [*file specifier*]
PURGE ALL

Deletes a file (or files) from RAM.

```
PURGE "TESTFILE:PORT"    PURGE ALL
PURGE KEYS               PURGE STRESS7
PURGE "TO"               PURGE
```

PUT *key name*

Enters a key code (specified by *key name)* into the key buffer.

```
PUT A$          PUT "fQ"          PUT "#43"
```

PWIDTH *print width*

Defines the line length of **PRINT** and **PLIST** statements. The *print width* is evaluated as an integer in the range 1 through 255. Noninteger values are rounded to the nearest integer. Values less than 1 are interpreted as 1; values greater than 255 are interpreted as **INF**.

```
PWIDTH INF          PWIDTH 20
```

R

RADIANS

Sets the unit of measure for expressing angles to radians. It is a short form of the **OPTION ANGLE RADIANS** statement.

RANDOMIZE [*seed*]

Specifies a seed for the **RND** function. If a *seed* isn't specified, then the computer uses the value of the current clock setting.

```
RANDOMIZE          RANDOMIZE R
```

READ *variable list*

Assigns values from **DATA** statements to variables.

```
READ A,B,X(A)      READ Z$[3,3],C      READ S(),T(,)
```

READ # *channel* [, *record*]; *variable list*

Reads data items from a data file associated with a specified channel number.

```
READ #3;Q,R,D$,P(),S1$()      READ #H;Z(,)
READ #7,20;I,A(I),X
```


REAL *item* [, *item...*]

Allocates memory for real variables and arrays. An *item* can be: *numeric variable* [(*dimension 1* [, *dimension 2*])]

```
REAL X                    REAL Q4(7,3)
REAL N, P(4)
```

REM/! comments

Enables you to document your programs with comments.

```
REM Program to compute average speed
! Program to compute average speed
IF A THEN GOSUB 750 ! Subroutine for depth factor.
```

RENAME [*file specifier*] TO *file specifier*

Changes the name of the specified file. Default file is the current file.

```
RENAME TEST1:PORT(2) TO TEST2
RENAME KEYF32 TO KEYS
RENAME TO STRESS6
RENAME A$ TO B$
RENAME KEYS TO "KEYS"
```

RENUMBER [*new start* [, *increment* [, *old start* [, *old final*]]]]

Renumbers the program lines in the current file.

```
RENUMBER                  RENUMBER 10,5
RENUMBER 300,10,100
```

RESET
RESET CLOCK

RESET resets user and system flags to their default settings.

RESET CLOCK nullifies the effect of executing **EXACT**.

RESTORE [*statement identifier*]

Specifies which **DATA** statement will be used by the next **READ** operation. If a *statement identifier* isn't specified, the computer moves the **DATA** pointer to the first item in the first **DATA** statement in the program.

```
RESTORE                    RESTORE 100
RESTORE ARRAY2
```

RESTORE # *channel* [, *record*]

Sets the file pointer associated with the specified channel number to the indicated record number. The default *record* is 0.

```
RESTORE #2                 RESTORE #3,20
RESTORE #I,99999
```

RETURN

Returns program execution to the statement following the invoking **GOSUB**.

RUN [*line number*]
RUN [*file specifier*][, *statement identifier*]

Executes a BASIC or binary program.

```
RUN                        RUN 100
RUN :CARD                  RUN PROG1
RUN ,START5                RUN :CARD,30
RUN TEST3:PORT(1),3500     RUN ,L$
```

S

SCI *number of significant digits*

Sets the scientific display format and the *number of significant digits* to be displayed or printed.

```
SCI D                      IF X < 1 THEN SCI 11
```


SECURE [*file specifier*]

Protects a file from being altered or purged. The default file is the current file.

```
SECURE "TESTDVC:PORT(2)"
SECURE KEYS
```

SETDATE *numeric date*
SETDATE *string date*

Sets the system clock to the specified date. The *numeric date* is of the form *YYDDD* where *YY* is two digits representing the year and *DDD* is three digits representing the number of the day in that year. The *string date* is of the form *YY/MM/DD* where *YY* is two digits representing the year, *MM* is two digits representing the month, and *DD* is two digits representing the day of the month. Rather than using a two-digit year (*YY*), you can use a four-digit year (for example, 1984).

```
SETDATE "1984/11/05"       SETDATE 82188
SETDATE 1982305            SETDATE D
SETDATE "83/07/21"         SETDATE D$
```

If you use the two-digit year format:

- If 60 <= *YY* <= 99, then *year* = *YY* + 1900.
- If 0 <= *YY* <= 59, then *year* = *YY* + 2000.

SETTIME *seconds since midnight*
SETTIME *time string*

Sets the time on the system clock.

```
SETTIME 3*60*60 + 15*60 + 12
SETTIME T
SETTIME 100.34
SETTIME T$
SETTIME "15:30:17"
```

SFLAG *flag number* [, *flag number...*]
SFLAG ALL/MATH

Sets the flag specified by each *flag number*, the math exception flags, or all user flags and settable system flags.

SFLAG sets flags as follows:

- **SFLAG ALL** sets all user flags (0 through 63).
- **SFLAG MATH** sets the math exception flags.

SFLAG sets specified user and system flags. A *flag number* is a numeric expression whose integer-rounded value represents a flag number.

SFLAG cannot set system flags numbered less than -32.

```
SFLAG ALL                SFLAG MATH
SFLAG 1,INX,OVF
```

SHORT *item* [, *item...*]

Allocates memory for short (5-digit precision) variables and arrays. An *item* can be: *numeric variable* [(*dimension 1* [, *dimension 2*])]

```
SHORT T                  SHORT R7(7,3)
SHORT X2, M(4)
```

SHOW[:]PORT

Displays the device type number (shown in the table below) and size, in bytes, of all plug-in memory devices and independent RAM currently in your HP-71. (*Not programmable.*)

```
SHOW PORT                SHOW :PORT
```

Device Type	Device Type Number
Independent Ram	1
ROM	2

STARTUP *command string*

STARTUP defines a command string to be executed when you turn on the HP-71.

```
STARTUP "RUN MONITOR"
STARTUP ""
STARTUP "WIDTH INF @ DELAY .5, INF"
```

STAT *array name* [(*number of variables*)]

Either creates and dimensions a statistical array, specified by *array name*, to the appropriate size for a specified *number of variables*, or designates a previously dimensioned statistical array as the current statistical array.

```
STAT A(4)                STAT B
```

STD

Selects the HP-71's standard BASIC format for displaying numbers.

STOP

Ends a subprogram, user-defined function, or a program.

SUB *subprogram name* [(*formal parameters*)]

SUB is the first statement in a subprogram and can specify the subprogram's formal parameters.

```
SUB PLOT(X,Y,D())        SUB A(N,B$,#8)
SUB PAYROLL
```

T

TRACE FLOW/VARS/OFF

Traces changes in variables and program flow for debugging.

- **TRACE FLOW** reports changes in the flow of a running program.
- **TRACE VARS** reports all variable assignments in a running program.
- **TRACE OFF** turns off all **TRACE** operations.

TRACE VARS and **TRACE FLOW** can be active at the same time.

TRANSFORM [*source file*] INTO *file type* [*destination file*]

Transforms BASIC program files into TEXT files and TEXT files into BASIC files for interchange purposes.

```
TRANSFORM LIF1PROG:PORT INTO BASIC PROG
TRANSFORM A$ INTO BASIC
TRANSFORM INTO TEXT
```

U

UNPROTECT

Removes the write protection from one track of a magnetic card.

UNSECURE [*file specifier*]

Clears the file access restriction that is set by **SECURE**.

```
UNSECURE KEYS
UNSECURE CAT$(N)
UNSECURE "TESTFILE:PORT(2)"
UNSECURE FILE2
UNSECURE
```


USER [ON/OFF]

Activates or deactivates the set of current key assignments.

- **USER ON** activates the User keyboard.
- **USER OFF** deactivates the User keyboard.
- **USER** switches the setting of the User keyboard.

W

WAIT *seconds*

Causes the computer to wait for the specified number of seconds.

```
WAIT 5          WAIT X          WAIT .3
```

WIDTH *display width*

Defines the line length for **DISP** and **LIST**.

```
WIDTH INF          WIDTH 20
```

WINDOW *first column* [*last column*]

Sets the display window size and location.

```
WINDOW 5,20          WINDOW 1
```

@

Joins statements together.

```
A=B @ USER ON @ S=COS(c)
```

Immediate Execute Keystrokes

For the following keystrokes, an asterisk indicates that holding down the key for more than about one-half second repeats the key's action.

[f] [SST] — Displays and executes the next program statement.

[f] [BACK] * — Moves the cursor one position to the left and erases the character in that position.

[f] [-CHAR] * — Erases the character under the cursor; fills the resulting gap by moving all characters to the right of the cursor one space to the left.

[f] [I/R] * — Switches between the Insert (▌) and Replace (▉) cursors.

[f] [LC] — Switches between uppercase and lowercase for the primary and g-shifted functions of the letter keys.

[f] [-LINE] — When editing a line, erases the character under the cursor and all characters to the right of the cursor.

During execution of ***CAT ALL*** and ***CAT :PORT***, pressing [f] [-LINE] begins the catalog display of the next higher-numbered port.

[f] [USER] — Activates or deactivates the User keyboard.

[f] [VIEW] — The next shifted or unshifted key pressed after [f] [VIEW] displays, while held down, its user-defined key definition. If the key has not been redefined, the HP-71 displays ***Unassigned***.

[f] [CALC] — Switches between BASIC and CALC modes.

[f] [CONT] — Continues a suspended program.

[RUN] — Runs the program in the current file.

[◀] * — Moves the cursor one space to the left without erasing a character. If part of the line is out of the display window to the left and the cursor is at the left edge of the window, [◀] scrolls the window to the left.

[▶] * — Similar to [◀], except moves the cursor to the right.

[▲] * — Displays the program line preceding the current line for editing and makes it the current line.

If the HP-71 is in the Command Stack, [▲] displays for editing the previously entered line in the stack.

If the HP-71 is displaying a multifile catalog, [▲] displays the preceding file catalog.

[▼] * — Displays the program line following the current line for editing and makes it the current line.

If the HP-71 is in the Command Stack, [▼] displays for editing the succeeding line in the stack.

If the HP-71 is displaying a multifile catalog, [▼] displays the succeeding file catalog.

[END LINE] * — Enters the displayed line into the computer for processing.

[ATTN] — Program not running: Clears the display.

Program running: Suspends the program (**SUSP** annunciator turns on) and clears the display.

If an entry is being made in response to an **INPUT** statement but [END LINE] has not yet been pressed, [ATTN] clears the typed entry, allowing another entry to be typed. Also, if you press [ATTN] twice, the HP-71 clears the entry, pauses the program, and clears the display.

g CTRL	Causes the next key pressed to be interpreted as the control function of that key. For the characters or actions associated with the control operation of HP-71 keys, refer to "HP-71 ASCII Character Set and Character Codes," page 41.
g ◀|	Moves the cursor to the first character in the line.
g |▶	Moves the cursor to the character position immediately to the right of the last character in the line.
g ERRM	While held down, displays the last error or warning message.
g ▲	Displays for editing the lowest-numbered program line or, if the command stack is active, the oldest command stack line. If you are displaying file catalogs, displays the catalog of the oldest file in a memory device.
g ▼	Displays for editing the highest-numbered line or, if the command stack is active the latest command stack line. If you are displaying file catalogs, displays the catalog of the newest file in a memory device.
g CMDS *	Activates or deactivates the command stack.
g 1 USER	Activates or deactivates the User keyboard for the next unshifted or shifted keystroke only.
f EDIT	When displaying a multifile catalog, makes the displayed file the current file.
ON /	When pressed simultaneously, interrupts any HP-71 operation and enables you to perform an ***INIT: 1***, ***INIT: 2***, or ***INIT: 3*** initialization. ***INIT: 1*** ends program execution and returns the BASIC prompt. ***INIT: 2*** ends program execution and tests the internal ROMs. ***INIT: 3*** causes a memory reset, destroying all files in main RAM.

HP-71 Character Set and Character Codes

The characters with character codes in the range of 0 through 127 and the default characters with character codes in the range of 128 through 255 are, in most cases, identical. Where there is a difference between the characters for the corresponding codes, the character in the lower range is a blank and the character in the upper range is shown in parentheses. (To represent a character using a code in the upper range, simply add 128 to the code shown for that character in the following table. Similarly, to convert a binary code in the table to the corresponding binary code for the upper range, replace the leftmost "0" in the tabular code with a "1".)

Note: You can use the **CHARSET** statement to change the display character symbol for one or more of the default display characters corresponding to character codes 128 through 255.

Hex	Binary	Decimal (CHR$) Character Code	Display Character	Keystrokes
00	0000 0000	0	none	g CTRL g @
01	0000 0001	1	▫	g CTRL A
02	0000 0010	2	⊠	g CTRL B
03	0000 0011	3	÷	g CTRL C
04	0000 0100	4	α	g CTRL D
05	0000 0101	5	β	g CTRL E
06	0000 0110	6	Γ	g CTRL F
07	0000 0111	7	♣	g CTRL G
08	0000 1000	8	(↹)	g CTRL H
09	0000 1001	9	σ	g CTRL I
0A	0000 1010	10	(↳)	-None-
0B	0000 1011	11	∩	g CTRL K
0C	0000 1100	12	μ	g CTRL L

Hex	Binary	Decimal (CHR$) Character Code	Display Character	Keystrokes
0D	0000 1101	13	(↵)	g CTRL M
0E	0000 1110	14	♫	g CTRL N
0F	0000 1111	15	☼	g CTRL O
10	0001 0000	16	►	g CTRL P
11	0001 0001	17	◄	g CTRL Q
12	0001 0010	18	↕	g CTRL R
13	0001 0011	19	‼	g CTRL S
14	0001 0100	20	¶	g CTRL T
15	0001 0101	21	§	g CTRL U
16	0001 0110	22	▬	g CTRL V
17	0001 0111	23	↨	g CTRL W
18	0001 1000	24	↑	g CTRL X
19	0001 1001	25	↓	g CTRL Y
1A	0001 1010	26	→	g CTRL Z
1B	0001 1011	27	(←)	g CTRL g [
1C	0001 1100	28	∑	-None-
1D	0001 1101	29	≠	g CTRL g]
1E	0001 1110	30	£	g CTRL g ^
1F	0001 1111	31	▒	-None-
20	0010 0000	32	Space	SPC
21	0010 0001	33	!	g !
22	0010 0010	34	"	g "
23	0010 0011	35	#	g #
24	0010 0100	36	$	g $
25	0010 0101	37	%	g %
26	0010 0110	38	&	g &
27	0010 0111	39	'	g '
28	0010 1000	40	(	(
29	0010 1001	41	)	)
2A	0010 1010	42	*	*

Hex	Binary	Decimal (CHR$) Character Code	Display Character	Keystrokes
2B	0010 1011	43	+	+
2C	0010 1100	44	,	,
2D	0010 1101	45	-	-
2E	0010 1110	46	.	.
2F	0010 1111	47	/	/
30	0011 0000	48	0	0
31	0011 0001	49	1	1
32	0011 0010	50	2	2
33	0011 0011	51	3	3
34	0011 0100	52	4	4
35	0011 0101	53	5	5
36	0011 0110	54	6	6
37	0011 0111	55	7	7
38	0011 1000	56	8	8
39	0011 1001	57	9	9
3A	0011 1010	58	:	g :
3B	0011 1011	59	;	g ;
3C	0011 1100	60	<	g <
3D	0011 1101	61	=	=
3E	0011 1110	62	>	g >
3F	0011 1111	63	?	g ?
40	0100 0000	64	@	g @
41	0100 0001	65	A	A
42	0100 0010	66	B	B
43	0100 0011	67	C	C
44	0100 0100	68	D	D
45	0100 0101	69	E	E
46	0100 0110	70	F	F
47	0100 0111	71	G	G
48	0100 1000	72	H	H

Hex	Binary	Decimal (CHR$) Character Code	Display Character	Keystrokes
49	0100 1001	73	I	I
4A	0100 1010	74	J	J
4B	0100 1011	75	K	K
4C	0100 1100	76	L	L
4D	0100 1101	77	M	M
4E	0100 1110	78	N	N
4F	0100 1111	79	O	O
50	0101 0000	80	P	P
51	0101 0001	81	Q	Q
52	0101 0010	82	R	R
53	0101 0011	83	S	S
54	0101 0100	84	T	T
55	0101 0101	85	U	U
56	0101 0110	86	V	V
57	0101 0111	87	W	W
58	0101 1000	88	X	X
59	0101 1001	89	Y	Y
5A	0101 1010	90	Z	Z
5B	0101 1011	91	[	g [
5C	0101 1100	92	\	-None-
5D	0101 1101	93	]	g]
5E	0101 1110	94	^	g ^
5F	0101 1111	95	_	-None-
60	0110 0000	96	`	-None-
61	0110 0001	97	a	g a
62	0110 0010	98	b	g b
63	0110 0011	99	c	g c
64	0110 0100	100	d	g d
65	0110 0101	101	e	g e
66	0110 0110	102	f	g f

Hex	Binary	Decimal (CHR$) Character Code	Display Character	Keystrokes
67	0110 0111	103	g	g g
68	0110 1000	104	h	g h
69	0110 1001	105	i	g i
6A	0110 1010	106	j	g j
6B	0110 1011	107	k	g k
6C	0110 1100	108	l	g l
6D	0110 1101	109	m	g m
6E	0110 1110	110	n	g n
6F	0110 1111	111	o	g o
70	0111 0000	112	p	g p
71	0111 0001	113	q	g q
72	0111 0010	114	r	g r
73	0111 0011	115	s	g s
74	0111 0100	116	t	g t
75	0111 0101	117	u	g u
76	0111 0110	118	v	g v
77	0111 0111	119	w	g w
78	0111 1000	120	x	g x
79	0111 1001	121	y	g y
7A	0111 1010	122	z	g z
7B	0111 1011	123	{	g {
7C	0111 1100	124	|	-None-
7D	0111 1101	125	}	g }
7E	0111 1110	126	~	-None-
7F	0111 1111	127	⊢	-None-

Key Identification Numbers

Notes: 1. These numbers assume uppercase is set.
2. Numbers for [f] and [g] are not useable.

Unshifted Keys

Key Code	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Key	Q	W	E	R	T	Y	U	I	O	P	7	8	9	/
Key Code	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Key	A	S	D	F	G	H	J	K	L	=	4	5	6	*
Key Code	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Key	Z	X	C	V	B	N	M	(	)	END LINE	1	2	3	–
Key Code	43			46	47	48	49	50	51		53	54	55	56
Key	ON	f	g	RUN	◄	►	SPC	▲	▼		0	.	,	+

[f] Shifted Keys

Key Code	57	58	59	60	61	62	63	64	65	66	67	68	69	70
Key	IF	THEN	ELSE	FOR	TO	NEXT	DEF	KEY	ADD	LR	PREDV	MEAN	SDEV	SQR
Key Code	71	72	73	74	75	76	77	78	79	80	81	82	83	84
Key	CALL	GOSUB	RETURN	GOTO	INPUT	PRINT	DISP	DIM	BEEP	FACT	SIN	COS	TAN	EXP
Key Code	85	86	87	88	89	90	91	92	93	94	95	96	97	98
Key	EDIT	CAT	NAME	PURGE	FETCH	LIST	DELETE	AUTO	COPY	RES	ASIN	ACOS	ATAN	LOG
Key Code	99			102	103	104	105	106	107		109	110	111	112
Key	OFF			SST	BACK	–CHAR	I/R	LC	–LINE		USER	VIEW	CALC	CONT

[g] Shifted Keys

Key Code	113	114	115	116	117	118	119	120	121	122	123	124	125	126
Key	q	w	e	r	t	y	u	i	o	p	'	¦	¦	^
Key Code	127	128	129	130	131	132	133	134	135	136	137	138	139	140
Key	a	s	d	f	g	h	j	k	l	;	$	%	&	:
Key Code	141	142	143	144	145	146	147	148	149	150	151	152	153	154
Key	z	x	c	v	b	n	m	[	]	CMDS	!	"	#	@
Key Code	155			158	159	160	161	162	163		165	166	167	168
Key				CTRL	⏮	⏭	ERRM	⏫	⏬		1 USER	<	>	?

System Flags

Flag Number	Effect When Set	Set/Clear by User
-1	Warning messages suppressed.	Yes
-2	Beeper is off.	Yes
-3	Continuous on.	Yes
-4	Inexact result (**INX**).	Yes
-5	Underflow (**UNF**).	Yes
-6	Overflow (**OVF**).	Yes
-7	Division by zero (**DVZ**).	Yes
-8	Invalid operation (**IVL**).	Yes
-9	User keyboard is active.	Yes
-10	Angular setting is Radians	Yes
-11, -12	Round-off setting.	Yes
-13, -14	Display format.	Yes
-15	Lower case lock.	Yes
-16	Base option 1.	Yes
-17 to -20	Number of display digits.	Yes
-25	Beep set to loud.	Yes
-26	Suppresses display of BASIC prompt	Yes
-46	Exact flag.	No
-57	**AC** annunciator on.	No
-60	Alarm annunciator (**((•))**) on.	No
-61	**BAT** annunciator on.	No
-62	**PRGM** annunciator on.	No
-63	**SUSP** annunciator on.	No
-64	**CALC** annunciator on.	No

Keyword Index

This index lists the HP-71 keywords by category and gives a page number where that keyword is introduced in this guide. Some keywords appear in more than one category.

Program Entry/Editing

Program Execution

Program Control

Debugging

Storage Allocation

Logical and Relational Operators

Arithmetic Operators

General Math

Logarithmic Operations

Trigonometric Operations

Statistics

Constants

Strings

Input/Output

Graphics

File Management

Time and Date

System Settings and Flags

Customization and Keyboard Control